APPENDIX A

```
/*
 * Domain.java
 *
 * Copyright (c) 2000 - General Magic, Inc. - All rights reserved
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION OF GENERAL MAGIC
 * Use and reproduction restricted by U.S. copyright law and
 * your contract with General Magic.
 *
 */
package com.genmagic.util;

public interface Domain {
    public boolean isVisited();
    public boolean setVisited(boolean b);
    public String getName();
}
```

```
/*
 * DomainController.java
 *
 * Copyright (c) 2000 - General Magic, Inc. - All rights reserved
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION OF GENERAL MAGIC
 * Use and reproduction restricted by U.S. copyright law and
 * your contract with General Magic.
 *
 */
package com.genmagic.util;

import java.util.Vector;

public class DomainController implements java.io.Serializable {
    Vector _domains = new Vector();
    int _current = 0;

    public DomainController() {
    }

    public void setDomains(Vector v) {
        _domains = v;
    }

    private Domain getCurrentDomain() {
        return getDomainAt(_current);
    }

    private Domain getDomainAt(int i) {
        return (Domain)_domains.elementAt(i);
```

```
}

public boolean isCurrentDomainVisited() {
    return getCurrentDomain().isVisited();
}

public String getDomainName() {
    return getCurrentDomain().getName();
}

public void moveToNext() {
    for (;_current < _domains.size()-1; _current ++) {
        if (getCurrentDomain().isVisited() == false) {
            getCurrentDomain().setVisited(true);
            return;
        }
    }
}

public void moveToPrevious() {
    _current --;
}

public boolean hasPrevious() {
    return _current >0;
}

public boolean hasMore() {
    for (int i=_current;i < _domains.size()-1; i ++) {
        if (getDomainAt(i).isVisited() == false) {
            return true;
```

```
            }
        }
        return false;
    }
}
```

```
<?xml version="1.0"?>
<!--
 * FILE_NAME
 *
 * Copyright (c) 2000 - General Magic, Inc. - All rights reserved
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION OF GENERAL MAGIC
 * Use and reproduction restricted by U.S. copyright law and
 * your contract with General Magic.
 *
 -->

<% String DNIS = request.getParameter("DNIS");
   String ANI = request.getParameter("ANI"); %>
<vxml version="1.0">

  <form>
    <block>
     <% if (DNIS.startsWith("800123")) { %>
       <!-- rout a call to banking service number to banking application -->
       <goto next="http://banking.genmagic.com/bankingService.jsp"/>
     <% if (ANI.startsWith("8004567890")) { %>
       <!-- rout a call from specific number to specific URL -->
       <goto next="http://my.genmagic.com/oshima.jsp"/>
     <% } %>
    </block>
  </form>
</vxml>
```

```
/*
 * PromptSet.java
 *
 * Copyright (c) 2000 - General Magic, Inc. - All rights reserved
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION OF GENERAL MAGIC
 * Use and reproduction restricted by U.S. copyright law and
 * your contract with General Magic.
 *
 */
package com.genmagic.util;

import java.util.Vector;
import java.util.Enumeration;

abstract public class PromptSet implements java.io.Serializable {

    protected Vector _prompts = new Vector();
    transient private Vector _cache = null;

    synchronized public void setAddPrompt(String promptname) {
        _cache = null;
        _prompts.addElement(promptname);
    }

    synchronized public void addPrompts(String []prompts) {
        _cache = null;
        for (int i=0; i<prompts.length; i++) {
            _prompts.addElement(prompts[i]);
        }
    }
```

```
    public String getPrompts() {
        return toVXML();
    }

    abstract protected Vector construct();

    protected static java.util.Random rand = new java.util.Random();

    public String toVXML() {
        if (_cache == null) {
            synchronized (this) {
                if (_cache == null) {
                    _cache = construct();
                }
            }
        }
        return (String)_cache.elementAt( (int)(rand.nextFloat() * _cache.size()));
    }
}
```

```
/*
 * Message.java
 *
 * Copyright (c) 2000 - General Magic, Inc. - All rights reserved
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION OF GENERAL MAGIC
 * Use and reproduction restricted by U.S. copyright law and
 * your contract with General Magic.
 *
 */
package com.genmagic.util;

import java.util.Date;

public interface Message {
    public String getForm();
    public String getSubject();
    public Date getDate();
    public String getBody();
}
```

```
/*
 * EmailList.java
 *
 * Copyright (c) 2000 - General Magic, Inc. - All rights reserved
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION OF GENERAL MAGIC
 * Use and reproduction restricted by U.S. copyright law and
 * your contract with General Magic.
 *
 */
package com.genmagic.util;

import java.util.Vector;
import java.util.Enumeration;

public class EmailList implements java.io.Serializable {
    Vector _messages = new Vector();
    int _current = 0;
    public EmailList() {
    }

    public void setMessages(Vector v) {
        _messages = v;
    }

    public Message getCurrentMessage() {
        return (Message)_messages.elementAt(_current);
    }

    public int getNumberOfMessage() {
        return _messages.size();
```

```
    }

    public void moveToNext() {
        _current ++;
    }

    public void moveToFirst() {
        _current = 0;
    }

    public void moveToLast() {
        _current = _messages.size() - 1;
    }

    public void moveToPrevious() {
        _current --;
    }

    public boolean isLast() {
        return _current == _messages.size() -1;
    }

    public boolean isFirst() {
        return _current == 0;
    }
}
```

```
/*
 * RandomPrompts.java
 *
 * Copyright (c) 2000 - General Magic, Inc. - All rights reserved
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION OF GENERAL MAGIC
 * Use and reproduction restricted by U.S. copyright law and
 * your contract with General Magic.
 *
 */
package com.genmagic.util;

import java.util.Vector;
import java.util.Enumeration;

public class RandomPrompts extends PromptSet {

    synchronized protected Vector construct() {
        int size = _prompts.size();
        Vector cache = new Vector();

        int base = 0;

        int six  = size % 6;
        int five = size % 5;
        int four = size % 4;

        int min = Math.min(six, five);
        int max = Math.max(six, five);

        if (size <= 8) {
```

```
    min = Math.min(min, four);
    max = Math.max(max, four);
}

if (min == 0) {
    if (min == six) {
        base = 6;
    } else if (min == five) {
        base = 5;
    } else if (min == four) {
        base = 4;
    }
} else {
    if (max == six) {
        base = 5;
    } else if (max == five) {
        base = 4;
    } else if (max == four) {
        base = 3;
    }
}

int rem = size % base;
int num = (size - rem) / base;

for (int i=0; i<size; ) {
    int n = base;
    if (rem-->0) n++;

    StringBuffer buf = new StringBuffer();
```

```
        buf.append("<var name='tmp' expr='Math.random()*" +
                   n + "'/>\n");
        buf.append("<if cond='1.0>=tmp'>\n");
        buf.append("  <audio src='builtin:" + _prompts.elementAt(i) + "'/>\n");
        i++;
        for (int j=1; j < n && i< size; j++, i++) {
            buf.append("<elseif cond='" + (j+1) +".0>=tmp'/>\n");
            buf.append("  <audio src='builtin:" + _prompts.elementAt(i) + "'/>\n");
        }
        buf.append("</if>");
        cache.addElement(buf.toString());
    }
    if (cache.size()==0) {
        cache.addElement("");
    }
    return cache;
}

public static void main(String a[]) throws Exception {
    RandomPrompts r = new RandomPrompts();
    r.setAddPrompt("a.wav");
    r.setAddPrompt("b.wav");
    r.setAddPrompt("c.wav");
    r.setAddPrompt("d.wav");
    r.setAddPrompt("e.wav");
    r.setAddPrompt("f.wav");
    r.setAddPrompt("g.wav");

    r.toVXML();
    System.out.println(r.toVXML());
}
```

```
}
```